

Guia docent

270005 - PRO2 - Programació II

Última modificació: 06/02/2025

Unitat responsable: Facultat d'Informàtica de Barcelona
Unitat que imparteix: 723 - CS - Departament de Ciències de la Computació.
Titulació: GRAU EN ENGINYERIA INFORMÀTICA (Pla 2010). (Assignatura obligatòria).
Curs: 2024 **Crèdits ECTS:** 7.5 **Idiomes:** Català, Castellà

PROFESSORAT

Professorat responsable: M. LUISA BONET CARBONELL - PABLO FERNANDEZ DURAN

Altres: Primer quadrimestre:
M. LUISA BONET CARBONELL - 11, 12, 13, 41
JUAN LUIS ESTEBAN ÁNGELES - 13
BORJA VALLES FUENTE - 12, 13, 41, 42

CAPACITATS PRÈVIES

Tenir formació sobre les tècniques de programació imperativa basada en:

- instruccions bàsiques: assignació, alternativa, iteració
- accions i funcions; pas de paràmetres; recursivitat
- vectors i tuples; seqüències
- esquemes de recorregut i cerca
- algorismes fonamentals: cerca binària, ordenació de vectors, aritmètica de matrius

Conèixer bé almenys un llenguatge imperatiu, preferentment C++. Experiència en la posada en marxa de programes C++ en l'entorn Linux.

Capacitat per assimilar informació a partir d'un enunciat. Capacitat per argumentar sobre la correctesa d'un algorisme i per comparar solucions algorísmiques.

COMPETÈNCIES DE LA TITULACIÓ A LES QUALS CONTRIBUEIX L'ASSIGNATURA

Específiques:

CT1.1A. Demostrar coneixement i comprensió dels conceptes fonamentals de la programació i de l'estructura bàsica d'un computador. CEFB4. Coneixement dels fonaments de l'ús i de la programació dels computadores, dels sistemes operatius, de les bases de dades i, en general, dels programes informàtics amb aplicació a l'enginyeria.

CT1.1B. Demostrar coneixement i comprensió dels conceptes fonamentals de la programació i de l'estructura bàsica d'un computador. CEFB5. Coneixement de l'estructura, funcionament i interconnexió dels sistemes informàtics, i dels fonaments de la seva programació.

CT1.2B. Interpretar, seleccionar i valorar conceptes, teories, usos i desenvolupaments tecnològics relacionats amb la informàtica i la seva aplicació a partir dels fonaments matemàtics, estadístics i físics necessaris. CEFB2. Capacitat per a comprendre i dominar els fonaments físics i tecnològics de la informàtica: electromagnetisme, ones, teoria de circuits, electrònica i fotònica i la seva aplicació per a la resolució de problemes propis de l'enginyeria.

CT3.6. Demostrar coneixement de la dimensió ètica a l'empresa: la responsabilitat social i corporativa en general i, en particular, les responsabilitats civils i professionals de l'enginyer en informàtica.

CT4.1. Identificar les solucions algorísmiques més adequades per a resoldre problemes de dificultat mitjana.

CT4.2. Raonar sobre la correcció i l'eficiència d'una solució algorísmica.

CT5.1. Triar, combinar i explotar diferents paradigmes de programació, en el moment de construir software, tenint en compte criteris com la facilitat de desenvolupament, l'eficiència, la portabilitat i la mantenibilitat.

CT5.2. Conèixer, dissenyar i utilitzar de forma eficient els tipus i les estructures de dades més adients per a la resolució d'un problema.

CT5.3. Dissenyar, escriure, provar, depurar, documentar i mantenir codi en un llenguatge d'alt nivell per a resoldre problemes de programació aplicant esquemes algorísmics i utilitzant estructures de dades.

CT5.4. Dissenyar l'arquitectura dels programes utilitzant tècniques d'orientació a objectes, de modularització i d'especificació i implementació de tipus abstractes de dades.

CT8.6. Demostrar comprensió de la importància de la negociació, dels hàbits de treball efectius, del lideratge i de les habilitats de comunicació en tots els entorns de desenvolupament de software.

Genèriques:

G5. Ser capaç de treballar com a membre d'un equip, ja sigui com a un membre més, ja sigui realitzant tasques de direcció, amb la finalitat de contribuir a desenvolupar projectes d'una manera pragmàtica i amb sentit de la responsabilitat; assumir compromisos tenint en compte els recursos disponibles.

METODOLOGIES DOCENTS

El temari s'exposa de forma molt pràctica, a través de la presentació de molts exemples.

Les classes de teoria introdueixen els coneixements, tècniques i conceptes que es posen en pràctica en les classes de laboratori. També inclouen la presentació i discussió de les solucions d'un conjunt de problemes.

Les dues hores de classes de teoria es fan setmanalment. Les tres hores de classes de laboratori també es fan setmanalment.

La programació de la pràctica integra els coneixements i les competències de tot el curs, potser tret de l'últim contingut del temari (tipus de dades recursius), que s'avalua al segon examen de teoria.

OBJECTIUS D'APRENENTATGE DE L'ASSIGNATURA

1. Dissenyar una classe de dades amb una clara independència entre la seva especificació i la seva implementació. Justificar que l'única forma de crear, consultar o modificar un objecte d'una classe de dades sigui a través de les operacions de l'especificació de la mateixa.
2. Resoldre en C++ qualsevol exercici basat en l'aplicació d'un esquema algorísmic senzill sobre un vector format per objectes d'una classe de dades.
3. Donada una implementació per a una classe de dades senzilla, introduir millores a la seva representació i a les seves operacions.
4. Explicar les fases del disseny modular.
5. Identificar en un enunciat textual d'un problema les abstraccions de dades susceptibles de donar lloc a classes que permetin resoldre el problema. Comprovar si alguna de les abstraccions identificades ja han estat detectades en situacions anteriors per tal de reutilitzar la classe corresponent.
6. Dissenyar de forma individual un programa modular en C++ a partir de les abstraccions de dades extretes del enunciat d'un problema. Afegir o modificar alguna funcionalitat d'un programa modular en C++ donat.
7. Codificar un programa modular en C++ amb bon estil i de forma que d'altres programadors siguin capaços d'entendre què fa i treballar sobre ell. Produir la documentació associada a un programa modular en C++ de manera que s'afavoreixi el seu ús per qualsevol altre programador.
8. Preparar un programa en C++ que faci servir tipus simples i classes de dades, algunes d'elles predefinides i d'altres definides pel propi alumne, per poder ser executat. Això s'ha de poder fer de dues maneres: 1) a base de compilar i muntar manualment amb el comandament g++; i 2) amb un makefile.
9. Dissenyar entre un grup d'estudiants un programa modular en C++ i/o un conjunt de jocs de proves que tractin totes les funcionalitats del mateix. Depurar sistemàticament aquest programa, de forma que s'hi puguin eliminar en un temps raonable petits errors d'implementació.
10. Conèixer els tipus de dades utilitzats habitualment per representar i manejar estructures de dades lineals i la seva especificació. Dissenyar algorismes iteratius i recursius per resoldre problemes de cerca i recorregut en piles, cues i llistes, usant les operacions del tipus de dades corresponent i iteradors (quan sigui recomanable).
11. Conèixer tipus de dades utilitzats per representar i manejar estructures de dades arborescents i la seva especificació. Dissenyar algorismes recursius per resoldre problemes de cerca i recorregut en arbres binaris, n-aris i generals usant les operacions del tipus de dades corresponent.
12. Descriure els passos principals del disseny d'algorismes iteratius. Justificar la correctesa d'algorismes iteratius relativament senzills.
13. Descriure els passos principals del disseny d'algorismes recursius. Justificar la correctesa d'algorismes recursius relativament senzills.
14. Conèixer el concepte d'immersió d'una funció, i explicar la diferència entre immersions d'especificació i immersions d'eficiència. Conèixer els diferents tipus d'immersions d'especificació i els diferents tipus d'immersions d'eficiència.
15. Donat un algorisme recursiu senzill, determinar si existeix una manera simple d'obtenir un algorisme iteratiu equivalent, i si és així, escriure'l.
16. Determinar si el cost d'un algorisme iteratiu o recursiu senzill donat que treballi sobre vectors, piles, cues, llistes o arbres és lineal o si és quadràtic (suposant que el cost sigui d'un d'aquests dos tipus).
17. Determinar si es pot millorar l'eficiència d'un algorisme recursiu senzill donat i, en cas de que sigui possible, dissenyar un algorisme recursiu alternatiu més eficient usant immersions d'eficiència.
18. Determinar si es pot millorar l'eficiència d'un algorisme iteratiu senzill donat i, en cas de que sigui possible, dissenyar un algorisme iteratiu alternatiu més eficient.
19. Implementar una estructura de dades amb requeriments específics sobre les seves operacions i/o l'eficiència de les mateixes usant tipus de

HORES TOTALES DE DEDICACIÓ DE L'ESTUDIANTAT

Tipus	Hores	Percentatge
Hores aprenentatge autònom	102,0	55.28
Hores activitats dirigides	7,5	4.07
Hores grup gran	30,0	16.26
Hores grup petit	45,0	24.39

Dedicació total: 184.5 h

CONTINGUTS

Estructures de dades lineals

Descripció:

Piles, cues, llistes, maps i sets: especificació i ús (operacions de cerca i recorregut). Iteradors: definició i ús.

Estructures de dades arborescents

Descripció:

Arbres binaris.

Correctesa de programes iteratius

Descripció:

Invariant d'un bucle. Justificació de la correctesa d'algoritmes iteratius.

Programació recursiva i correctesa d'algoritmes recursius

Descripció:

Disseny inductiu d'algoritmes recursius. Justificació de la correctesa d'algoritmes recursius. Immersió (o generalització) d'una funció. Immersions d'especificació: per afebliment de la postcondició i per reforçament de la precondició. Relació entre algoritmes recursius lineals finals i algoritmes iteratius.

Millores d'eficiència en programes recursius i iteratius

Descripció:

Detecció de la repetició de càlculs en programes recursius i iteratius. Immersions d'eficiència: nous dades (paràmetres d'entrada) i/o resultats (valors retornats o paràmetres de sortida) en operacions recursives per millorar l'eficiència. Noves variables locals que usen els seus valors en anteriors iteracions per millorar l'eficiència d'operacions iteratives.

Disseny modular i disseny basat en objectes

Descripció:

Abstracció i la seva necessitat. Descomposició funcional i per dades. Mòduls. Ocultació de la informació. Encapsulat. Fases del disseny modular: distinció entre especificació i implementació. Tipus de mòduls i el seu ús. Biblioteques.

Principis bàsics de disseny basat en objectes: classes i objectes; camps i mètodes.

Implementació de dissenys modulars en C++. Compilació separada i muntatge. Depuració, prova i documentació de programes modulars.

Tipus recursius de dades

Descripció:

Introducció a l'ús de tipus recursius de dades. El constructor de tipus punter i la gestió de memòria dinàmica. Implementació d'estructures de dades enllaçades mitjançant tipus recursius de dades. Algoritmes iteratius i recursius per resoldre problemes de cerca i recorregut en estructures de dades enllaçades accedint directament a la representació basada en nodes y punters a nodes.

ACTIVITATS

Exercicis de repàs de C++.

Descripció:

Enllaç amb els continguts de PRO1.

Objectius específics:

2, 8

Dedicació: 6h

Aprenentatge autònom: 3h

Grup petit/Laboratori: 3h

Estructures de dades lineals.

Descripció:

Desenvolupament del tema corresponent de l'assignatura i exercicis de laboratori.

Objectius específics:

10

Dedicació: 17h

Aprenentatge autònom: 9h

Grup gran/Teoria: 2h

Grup petit/Laboratori: 6h

Programació recursiva

Descripció:

Desenvolupament del tema corresponent de l'assignatura.

Objectius específics:

13, 14, 15

Dedicació: 6h

Aprenentatge autònom: 3h

Grup gran/Teoria: 3h

Estructures de dades arborescents.

Descripció:

Desenvolupament del tema corresponent de l'assignatura i exercicis de laboratori.

Objectius específics:

11

Dedicació: 27h

Aprenentatge autònom: 16h

Grup gran/Teoria: 2h

Grup petit/Laboratori: 9h

Correctesa de programes iteratius

Descripció:

Desenvolupament del tema corresponent de l'assignatura.

Objectius específics:

12

Dedicació: 4h

Aprenentatge autònom: 2h

Grup gran/Teoria: 2h

Milliores d'eficiència en programes recursius i iteratius.

Descripció:

Desenvolupament del tema corresponent de l'assignatura.

Objectius específics:

12, 13, 14, 15, 16, 17, 18

Dedicació: 24h

Aprenentatge autònom: 12h

Grup gran/Teoria: 3h

Grup petit/Laboratori: 9h

Introducció al disseny modular i al disseny basat en objectes.

Descripció:

Desenvolupament del tema corresponent de l'assignatura.

Objectius específics:

1, 2, 4, 21

Dedicació: 8h

Aprenentatge autònom: 4h

Grup gran/Teoria: 4h

Especificació i ús de classes d'objectes en C++.

Descripció:

Exercicis del tema corresponent de l'assignatura.

Objectius específics:

1, 2, 21

Dedicació: 9h

Aprenentatge autònom: 6h

Grup petit/Laboratori: 3h

Implementació de classes d'objectes en C++.

Descripció:

Exercicis del tema corresponent de l'assignatura.

Objectius específics:

1, 3, 21

Dedicació: 6h

Aprenentatge autònom: 3h

Grup petit/Laboratori: 3h

Supervisió pràctica

Descripció:

Supervisió del disseny i implementació de la pràctica.

Objectius específics:

1, 3, 5, 6, 7, 8, 10, 11, 16, 17, 21

Dedicació: 9h

Aprenentatge autònom: 6h

Grup petit/Laboratori: 3h

Tipus recursius de dades.

Descripció:

Desenvolupament del tema corresponent de l'assignatura.

Objectius específics:

1, 3, 16, 17, 18, 19, 20

Dedicació: 32h 30m

Aprenentatge autònom: 16h

Grup gran/Teoria: 7h 30m

Grup petit/Laboratori: 9h

Repàs de teoria i problemes d'exàmens.

Descripció:

Els alumnes podran preguntar qüestions concretes sobre els temes donats a teoria.

Dedicació: 4h

Aprenentatge autònom: 2h

Grup gran/Teoria: 2h

EX_SIMUL1

Descripció:

Examen de simulacre per al Parcial 1

Objectius específics:

2, 3, 5, 6, 7, 8, 10, 11, 16, 17, 18, 19, 20

Dedicació: 4h

Aprenentatge autònom: 2h

Activitats dirigides: 2h

EX_PAR1_TP

Descripció:

Primer examen parcial de teoria i problemes.

Objectius específics:

2, 3, 5, 6, 7, 8, 10, 11, 16, 17, 18, 19, 20

Dedicació: 7h

Aprenentatge autònom: 5h

Activitats dirigides: 2h

CODI_DOC_PRAC

Descripció:

Lliurament del codi i la documentació de la pràctica.

Objectius específics:

1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 21

Competències relacionades:

G5. Ser capaç de treballar com a membre d'un equip, ja sigui com a un membre més, ja sigui realitzant tasques de direcció, amb la finalitat de contribuir a desenvolupar projectes d'una manera pragmàtica i amb sentit de la responsabilitat; assumir compromisos tenint en compte els recursos disponibles.

Dedicació: 10h

Aprenentatge autònom: 10h

EX_SIMUL2

Descripció:

Examen de simulacre per al Parcial 2

Objectius específics:

2, 3, 5, 6, 7, 8, 10, 11, 16, 17, 18, 19, 20

Dedicació: 5h

Aprenentatge autònom: 2h 30m

Activitats dirigides: 2h 30m



EX_PAR2_TP

Descripció:

Segon examen parcial de teoria i problemes

Objectius específics:

2, 3, 5, 6, 7, 8, 10, 11, 16, 17, 18, 19, 20

Dedicació: 9h

Aprenentatge autònom: 6h 30m

Activitats dirigides: 2h 30m

SISTEMA DE QUALIFICACIÓ

La nota de les competències tècniques (NCTEC) es calcula de la manera següent:

$$\text{NCTEC} = 0.3 \cdot \text{EXAM1} + 0.3 \cdot \text{EXAM2} + 0.3 \cdot \text{PRAC} + 0.10 \cdot \text{DELIVERY}$$

on

* EXAM1 I EXAM2 són les notes del primer i segon exàmens parcials de teoria

* PRAC és la nota de la pràctica; podrà tenir un component obtingut automàticament mitjançant l'execució amb jocs de proves, i un component obtingut per correcció manual. L'EXAM2 tindrà un problema sobre la pràctica que farà mitjana amb aquesta.

* DELIVERY és la nota obtinguda de les entregues de resolució d'exercicis realitzades durant el curs.

No obstant, NCTEC serà NP si el pes dels actes avaluatoris amb nota NP és igual o superior al 70%.

L'avaluació de la competència transversal Treball en Equip es realitzarà a partir de l'enviament d'exercicis en grups petits d'estudiants.

REVALUACIÓ DE PRO2

Hi haurà reavaluació d'aquesta assignatura. Consisteix en un curs intensiu de 12 hores presencials amb la corresponent avaluació, que es fan un cop finalitzats els exàmens finals i abans de l'inici del quadrimestre següent. Es calcula que la reavaluació requereix unes 50 hores de dedicació per part de l'estudiant entre classes presencials, hores d'estudi personal i avaluació. Només hi poden optar els estudiants que compleixen determinats requisits. Les places són limitades i s'assignaran per ordre decreixent de nota. Cada estudiant podrà fer com a molt la reavaluació d'una de les assignatures de fase inicial que s'ofereixen.

Requisits mínims per optar a la reavaluació

Per a optar a la reavaluació és requisit indispensable estar matriculat de l'assignatura i haver obtingut una nota final d'entre 3.5 i 4.9. A més, cal que $0.3 \cdot \text{PRAC} + 0.10 \cdot \text{DELIVERY} \geq 2$.

Requisits per a ser avaluat del curs intensiu

Per tal de ser avaluat del curs intensiu és obligatori:

- Assistir a totes les classes presencials.
- Fer els exercicis o activitats que demani el professorat del curs.

Preinscripció i admissió

El procés d'inscripció es publicarà al racó. L'admissió i no assistència al curs pot comportar no ser admès en pròximes edicions.

Avaluació del curs de reavaluació

Es farà un examen de reavaluació (REVAL). El resultat de l'avaluació del curs intensiu serà Apte si $0.6 \cdot \text{REVAL} + 0.3 \cdot \text{PRAC} + 0.10 \cdot \text{DELIVERY} \geq 5$, i NoApte en cas contrari. La nota definitiva de l'assignatura serà:

NOTA_DEFINITIVA = 5, si la nota de l'intensiu és Apte;

NOTA_DEFINITIVA = NOTA_ASSIGNATURA, si la nota de l'intensiu és NoApte;

on NOTA_ASSIGNATURA és la nota sobre 10 obtinguda al curs quadrimestral.



BIBLIOGRAFIA

Bàsica:

- Alquezar, René; Valles, Borja; [et al.]. Apunts de teoria de PRO2.
- Valles, Borja; [et al.]. Sessions de Laboratori de PRO2.
- Cortadella, Jordi; Rubio, Albert; Valentin, Luis. Iniciació a la Programació (notes de curs). 2001.
- Weiss, M.A. Data structures and problem solving using C++. 2nd ed. Pearson Education International, 2003. ISBN 0321205006.
- Musser, D.R.; Derge, G.J.; Saini, A. STL tutorial and reference guide: C++ programming with the standard template library. 2nd ed. Addison-Wesley publishing company, 2000. ISBN 0201379236.

Complementària:

- Castro, J. [et al.]. Curs de programació. McGraw-Hill, 1992. ISBN 844810031X.
- Balcazar, J.L. Programación metódica. McGraw-Hill, 1993. ISBN 8448119576.
- Savitch, Walter J; Mock, Kenrick. Problem solving with C++. Tenth edition. Pearson, [2018]. ISBN 9780134448282.

RECURSOS

Enllaç web:

- <http://c.conclase.net/curso/index.php>. Curs de C++. Ideal com a manual de referència.
- <http://www.cplusplus.com/reference/stl/>. Manual de referència sobre els containers de la llibreria STL de C++.
- <https://pro2.cs.upc.edu>. Pàgina web dels professors de l'assignatura. Conté tota la documentació rellevant: apunts, col·leccions d'exercicis, enunciat de la pràctica.